Week-1 Assignment

Part 1

Q1

Groth16 and PLONK are two types of zkSNARKs

Q2

SNARK proofs rely on elliptical curves to derive proofs. Hence require public parameters that need to be derived.

The derivation of these public parameters is the reason why a trusted-setup is required. STARKs on the other hand rely on collision resistant hash functions to generate proofs eliminating the need for a trusted setup.

Q3

a. SNARKs are unbeaten in their proof size and verification time which is marginally lower than that of STARKs

b. SNARKs use elliptical curve cryptography to generate proofs while STARKs depend on collision resistant hashes making them more secure from high computational attack vectors.

Part 2

Q2.1

The circuit inside HelloWorld.circom is responsible for checking that for

2 given inputs (a, b) their multiplication must equal c

Q2.2.

Power of tau is a cryptographic MPC (Multi party computation) ceremony used to generate public parameters for SNARK proofs and is considered the first phase only. The second phase is circuit specific

and much more computationally intensive. SNARKs rely on CRS (Common Reference Strings) to

generate proofs. These CRS must be generated by a trusted third-party. While discarding the

toxic waste created which could be used to generate fraudulent proofs. To make this process

decentralized the Powers of Tau ceremony was created which relies on multiple participants

to generate this CRS while ensuring that if even 1 destroys their waste the CRS is safe

Q2.3

Phase 1 of the trusted setup ceremony is general in nature and is called the Power of Tau ceremony. It is responsible for generating a general string that can be used by anyone or any circuit. The Second Phase is much more computationally intensive and is circuit specific\ i.e., needs to be done for each and every circuit. So, a general ceremony can be conducted by a protocol but the Phase 2 needs to be performed by anyone wanting to use zkProofs.

Q3.2

The error arises because the circuit in Multiplier3 uses non-quadratic equations. i.e., all the constraints are required to be in the form of a*b+c and any circuit violating this will fail

Q4.1

The process for compiling PLONK and Groth16 differs on their need for phase 2 of the ceremony in the case of

Groth16 after public parameter generation another phase of circuit specific generation is required which is skipped in PLONK as one trusted setup can universally verify any given circuit. In code the difference is seen when we skip the contribution line for straight verification with the PLONK setup.

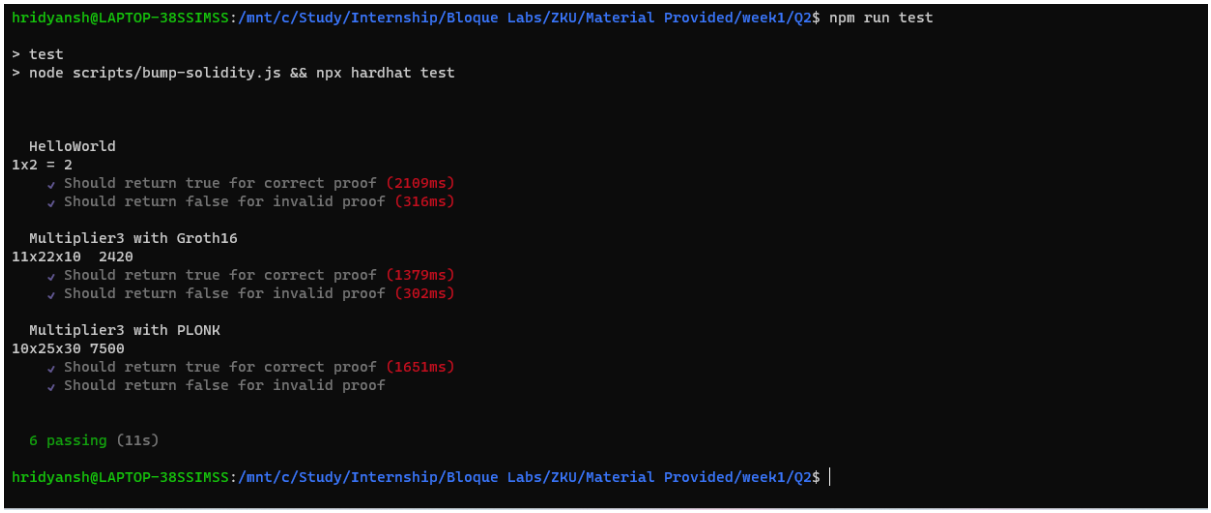
Q4.2

Practically the PLONK contract is much more complex but requires fewer inputs. While the Groth16 verifier requires much more carefully crafted inputs for a simpler contract.

Plonk takes a little more time in verifying as compared to Groth16. While it makes up for this by not having the need for Phase2 of the ceremony.

Q5.3

Test passing for Given Script



Part 3

Q1.1

I think that the number 32 tells the LessThan Component the greatest number to expect in bits.

So, for 32 it represents that the greatest number you can check is 2^32.

Q1.2

The output for the LessThan will be 1/0 depending on whether the condition is passing or failing. If given number is < 10 it will be 1 otherwise 0;

Q1.3

The first problem faced was that snarkjs was not installed and needed to be installed using npm

Q1.4

Solving the sudoku with a brute force method will require much more computational power and space

as compared to the algorithmic implementation making the circuit much more inefficient and harder to generate. Increasing the number of constraints and overall size of the circuit.